أكشن إسكربت 3.0

أساسيات الأكشن إسكربت

المحتويات الرئيسية المناقشة:

- لوحة محرر الأكشن إسكربت
- الشق التصميمي في برنامج الفلاش
- البرمجة الكائنية التوجه
- الخصائص
- الآليات
- الأحداث
- متفرقات

إسلام عبد الرحيم

بسم الله الرحمن الرحيم

لايوجد أدنى شك في أن الفلاش عالم كبير وأفكار كل يوم بتتولد وبتتجدد ولذلك كل شخص قرر يدخل هذا المجال فهو يعرف تماماً أنه لايوجد نهاية في طريق تعلم هذا البرنامج, وأكبر دليل على هذا أن هذا الكتاب لم يكن سوى مجرد أوراق مبعثرة كنت أدون فيها خلاصة ما أتعلمه ولكني أحببت أن أنشره لعله يجد طريقه إلى شخص يستفيد منه.

الأكشن إسكربت 3.0 جعل الشق البرمجي سهل جداً وأكثر تفاعلية وقوة وهذا كلام شركة أدوبي وصراحة بدايتي مع الفلاش بداءت مع الأكشن إسكربت 3.0 ولا أدري طبيعة الأكشن إسكربت 2.0 ولذلك لن أتكلم عنه في هذا الكتاب. هذه اللغة تحتاج فقط إلى التركيز والتأمل والتدريب. ربما تكون كلمة برمجة تحمل الكثير من الرهبة ولكن الأمر أسهل وأبسط بكثير مما تتخيل فالبرمجة ما هي إلا عملية تنظيم أو بالأحرى وضع نظام لشيء وهذا المخطط يبدأ بناءه خطوة خطوة وبالتدريج فهى مجرد قواعد ,أساسيات وأدوات تُوظفها كما تشاء, ولعلها من ألذ وأمتع لغات البرمجة بالإضافة إلى أن هناك أفكار كثيرة تتم من خلالها.

أحب أن أنوه فقط على أن كل ما هو مكتوب هو مجرد خلاصة ما فهمته وطبقته وليس بالضرورة الأسلوب الأمثل أو الأصح في توصيل الفكرة فأنا مازلت مبتدىء في تعلم هذه اللغة ولكني أسأل الله أن يستفيد منه كل من يطلع عليه. وأحب ايضاً أن أؤكد على أن كل الأمثلة المطروحة مراجعة بشكل دقيق ومُطبقة ومًنفذه وصحيحه مئة بالمئة.

إسلام عبد الرحيم

2007-9-10

مقدمة:

نشأة لغة الأكشن إسكربت...

كانت بداية برنامج الفلاش هى برنامج الـ FutureSplash Animator وقد تحوّل اسمه إلى Macromedia Flash عندما اشترته شركة ماكروميديا في شهر ديسمبر عام 1996. وقد مثّل هذا البرنامج بداية التطور المذهل في مجال تصميم مواقع الويب حيث أتاح لمصممي المواقع إضافة الحيوية والحركة والتفاعلية إليها والذي أدى بدوره إلى تراجع شعبية الصور المتحركة (جيف GIF) في ذلك الوقت, إلى أن قررت شركة ماكروميديا إضافة شق برمجي إلى برنامج الفلاش وكانت لغة الجافا إسكربت JavaScript الأكثر استخداماً في إضافة التفاعلية إلى صفحات الويب في ذلك الوقت ولذلك قررت شركة ماكروميديا بدلاً من أن تنشىء لغة جديدة مستقلة قررت أن توفّر لغة مبنية على نفس الأسس التي قامت عليها لغة الجافا إسكربت, فلغة الجافا إسكربت مشتقة من لغة إيكيما إسكربت ECMAScript.

ولذلك قامت شركة ماكروميديا عام 2000 بإدماج لغة الـ ActionScript المشتقة من لغة الإيكيما إسكربت مع الإصدار الـ 5 من برنامج الفلاش ليصبح برنامج متكامل ويبدأ رحلة التميّز, لذلك فلغة الأكشن إسكربت قريبة جداً ومشابهة للغة الجافا إسكربت.

الشق التصميمي لبرنامج الفلاش يتيح فقط عمل حركات بسيطة ومحدودة, بينما إضافة لغة الـ ActionScript أطلقت العنان لتنفيذ أي فكرة أو حركة قد تخطر بذهنك من أول إنشاء مواقع كاملة ببرنامج الفلاش إلى تصميم ألعاب وما بينهما من أفكار من تصميم اللافتات الإعلانية في مواقع الويب والبطاقات الإليكترونية وحتى العروض التقديمية. فحيثما تريد أن تذهب فسوف تأخذك هذه اللغة إلى ذاك المكان ..!

والجدير بالذكر أن شركة Adobe قامت بشراء برنامج الفلاش من شركة ماكروميديا في شهر ديسمبر 2005 وطرحت الإصدار الأخير وهو Adobe Flash CS3 الذي يدعم إستخدام لغة الـ ActionScript 3.0.

يبدو أن شهر ديسمبر الشهر الدائم لإنتقال ملكية برنامج الفلاش ..!

معلومة إضافية ..!

- اسم لغة الإيكيما إسكربت مشتق من European Computer Manufacturers Association وهى رابطة مصنعي الكمبيوتر الأوروبية, وهى رابطة متخصصة في وضع مقاييس مواصفات عدة لغات برمجية. و ECMA-262 من ضمن المواصفات التي قدمتها هذه الرابطة والمبني عليها الـ ECMAScript ومن ثم الـ JavaScript والـ ActionScript.
- المقطع CS3 في اسم آخر إصدار Adobe Flash CS3 هو اختصار لـ Creative Suite 3 أي حزمة الإبداع 3 وهى عبارة عن عدة برامج تصميم تشمل برنامج الفلاش وبرنامج الفوتوشوب والدريم ويفر وغيرها الكثير.

بعد هذه الجولة التاريخية, وكأي وافد جديد لابد وأن تتعرف عن المكان الذي سوف تعمل فيه سوياً مع هذه اللغة...!

لوحة محرر الأكشن إسكربت:

المُدمج داخل برنامج الفلاش...

هناك طريقتين للوصول إلى لوحة محرر الأكشن إسكربت Actions panel داخل برنامج الفلاش:

الأولى : عن طريق الذهاب إلى قائمة Window ثم إختيار Actions.

الثانية : عن طريق الضغط على مفتاح F9 من لوحة المفاتيح.

أي من الطريقتين استخدمت سوف تنتهي بفتح لوحة محرر الأكشن إسكربت, أنظر الصورة ..!

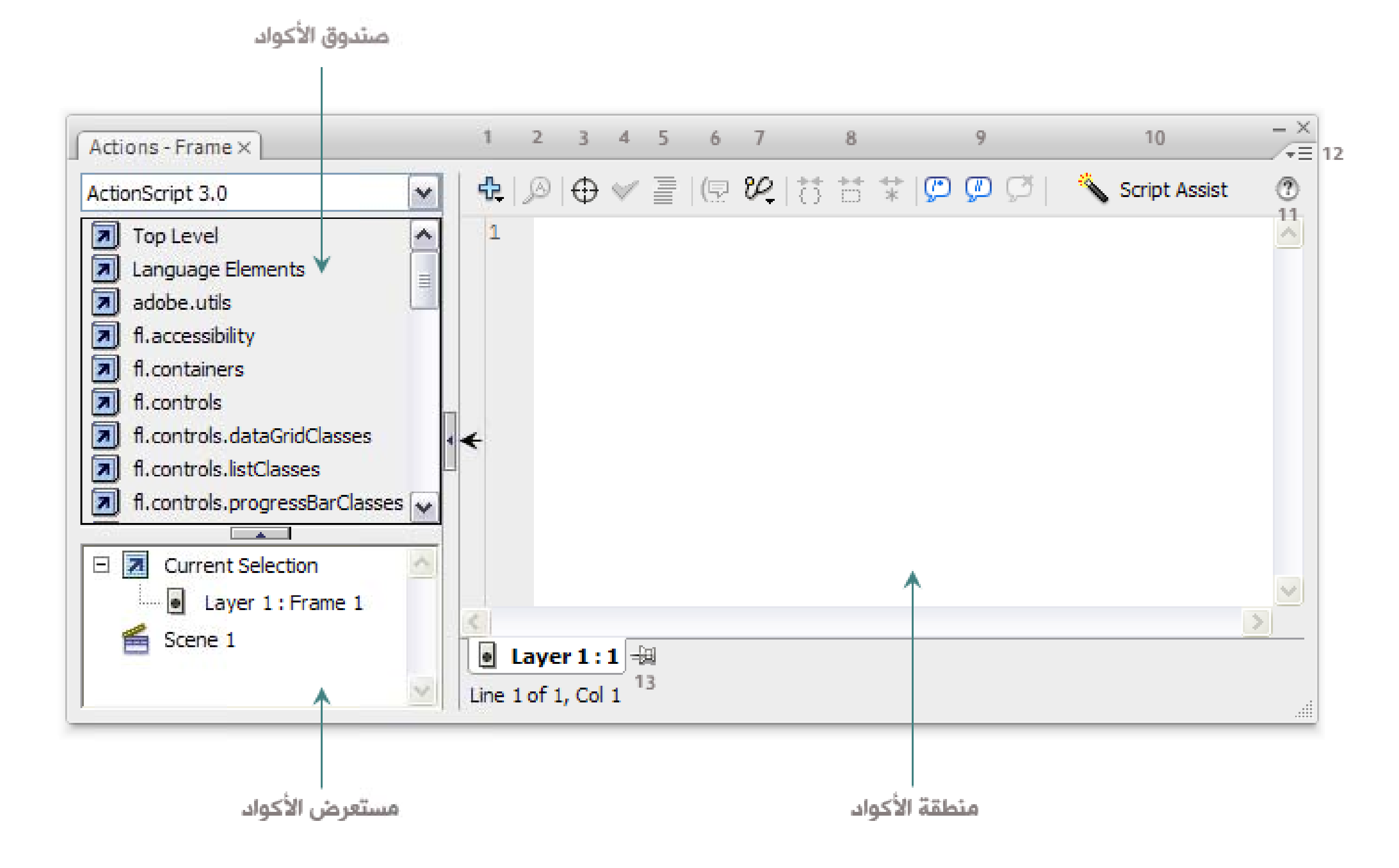
ويمكن تقسيمها إلى 3 أجزاء رئيسية:

1. صندوق الأكواد Actions toolbox:

يمكن تشبيهها بأنها مثل الكتالوج الذي يحوي جميع الأكواد (الدوال والخصائص) المتاحة في الـ ActionScript 3.0 التي يمكن استخدامها. فهى كالأرشيف, مصنفة ومقسمه تقسيم دقيق على حسب الفئة التي ينتمي إليها الكود ويمكنك البحث فيها عن كود محدد فكما تلاحظ وجود أيقونة على هيئة كتاب عليه سهم عند الضغط عليه يفتح قائمة بها كُتباً أخرى وهكذا إلى أن تصل إلى أيقونة على هيئة دائرة وعليها سهم وهذه تمثل الكود الذي يمكن استخدامه ويمكنك عن طريق اختياره والضغط عليه ضغطتين أن تضيفه إلى الكود الذي تكتبه أو عن طريق السحب والإفلات في منطقة الأكواد.

2. مستعرض الأكواد Script navigator:

أثناء كتابتك للبرمجة الخاصة بفيلم من الوارد أن تضيف أكواد إلى إطارات (لقطات) مختلفة. هذه المنطقة التي تسمى مستعرض الأكواد تقوم بعرض فقط الإطارات التي تحتوي على أكود, فتمكنك من التنقل بسهولة بين هذه الإطارت.



3. منطقة الأكواد Actions pane:

وهذه هي المنطقة الأهم التي سوف تكتب فيها الأكواد. تحتوي ايضاً هذه المنطقة على عدة أيقونات في الجزء العلوي وهي كالآتي:

- [1] تماماً مثل صندوق الأكواد يمكنك من خلالها البحث عن كود معين وإضافته.
- [2] أيقونة البحث عن كلمة في الكود الذي كتبته واستبدالها.
- [3] أيقونة مسار الكائن, فقد يحدث أن يكون لديك كائن داخل كائن أخر داخل أخر, فهذه الأيقونة تزودك بشجرة توضح علاقة الكائنات ببعضها فيمكنك اختيار كائن معين وهذه الايقونة تكوّن لك مساره.
- [4] أيقونة التدقيق النحوي والتي تتأكد من سلامة الأكود المكتوبة.
- [5] أيقونة التنسيق الذاتي, حيث تقوم بمحازاة الأكواد وتنسيقها.
- [6] أيقونة التلميحات, أثناء كتابة الكود يقوم المحرر بإظهار بعض التلميحات حول طريقة كتابة واستعمال دالة أو خاصية معينة.
- [7] أيقونة خيارات التصحيح, مثلاً إضافة أو إزالة نقاط توقف عند سطر معين للإختبار البرنامج.
- [8] أيقونات الإختزال, الأولى للإختزال سطر واحد بمعنى تجميعه وإختزاله إلى أيقونة صغيرة مُعلمة بعلامة [+] لتوفير مساحة, الثانية إختزال عدة أسطر, الثالثة للتراجع وعرض كل ما تم إختزاله.
- [9] أيقونات التعليقات, الأولى للإضافة تعليق عبارة عن عدة أسطر, الثانية للإضافة تعليق عبارة عن سطر واحد, الثالثة للإزالة التعليق, بعد تنشيطه.
- [10] أيقونة مساعد كتابة الأكواد وهي تقوم بإظهار صندوق فوق هذه الإيقونات وتُحّول المحرر إلى وضعية سطر بسطر (أي لا يمكنك كتابة أو إضافة أو تعديل أي شيء في منطقة كتابة الأكواد). فقط تحدد السطر ثم تضيف دالة أو خاصية من صندوق الأكواد وتملىء الفراغات الخاصة بالدالة أو الخاصية بما يناسبها من خلال صندوق مساعد كتابة الأكواد. وبالطبع أنت في غنى عن هذا المساعد كليةً ..!
- [11] أيقونة إظهار لوحة المساعدة.
- [12] قائمة محلية (خاصة بالمحرر فقط), وفيها أغلب الوظائف التي تقدمها الأيقونات السابقة بالإضافة إلى خيارات أخرى مثل تصدير الكود الحالي أو إستيراد كود أخر إلى الكود الحالى وخيار طباعته, السماح بالإلتفاف الكود (أي عدم جعل الكود في سطر واحد في حالة وجود سطر طويل), إظهار ترقيم الأسطر وهى خاصية مهمة جداً لأنه ربما يحدث أخطاء فيخبرك المحرر برقم السطر الذي يوجد به الخطأ, وهناك أيضاً نافذة التفضيلات التي تحدد فيها نوع الخط والألوان المستخدمة في كتابة الأكواد.

ملحوظة مهمة ..!

- قد ذكرت في وظيفة مستعرض الأكواد أنه يمكن من خلاله التنقل بين الإطارات التي تحتوي على أكواد ولكن ربما تود تثبيت إحداها وتفتح أكود إطار أخر من مستعرض الأكواد ولعمل ذلك, أفتح الإطار الذي تريد تثبيته واضغط على الزر رقم [13] في الصورة, لاحظ أن المحرر قام بعمل تبويب جديد, أفتح فيه ما تشاء ..!
- كلٍ من صندوق الأكواد ومستعرض الأكواد فالغالب لن تستخدم أي منهما ولذلك يُفضل أخفائهما عن طريق الضغط على الزر الذي يوجد عند منتصف الحاجز الذي يفصلهما عن منطقة كتابة الأكواد..! (أنظر الصورة, الزر مشار إليه بالسهم الأسود)

كل ما أريده منك من كل ما سبق هو أن تعرف فقط منطقة الأكواد, المكان الذي سوف تكتب فيه الأكواد. أما الباقي فيرجع لك أن أردت أن تعرفه الآن. لأنك سوف تتعرف وتتعود عليه مع الممارسة, لذلك لا تقلق ..!

الشق التصميمي في برنامج الفلاش:

علاقته بالأكشن إسكربت ..

هل فكرت يوماً كيف تم عمل أي فيلم كرتون مثل توم آند جيري ..؟!

الفكرة ببساطة أن هناك مجموعة عمل حيث يكون هناك الرسامين الرئيسين وأخرين مساعدين. أما الرسامون الرئسيون فيقومون برسم حركات البداية والنهاية الرئيسية, الـ Keyframes, مثلاً رسم القط أعلى الشجرة, ثم في رسمة أخرى القط وهو على الأرض. أما باقية اللقطات البينية, الـ inbetweens, التي تُمثل حالة القط وهو في الهواء وكيفية القفزة فيتركوها للرسامين المساعدين. ثم بعد ذلك تُجمّع بالترتيب وتُعرض بسرعة معينة مثلاً 12 لقطة في الثانية فتعطي لك تخيل بالحركة.

وهذه تماماً كانت فكرة برنامج الفلاش عندما خرج للوجود. وهى أنك تعطيه مجموعة لقطات مختلفة, متدرجة ومتتابعة ثم يقوم البرنامج بعرض مجموعة اللقطات بسرعة معينة تعطي تصور بالحركة. ولذلك كانت وحدة البناء الرئيسية في البرنامج هى اللقطة, Frame. ولذلك كان برنامج الفلاش لا يعرف ماذا ترسم أو علاقة عناصر اللقطة ببعضها, إلى أن قُدمت فكرة الرموز, الـ Symbols. ولنبدأ أولاً بتعريف كلمة رمز ..؟!

كلمة رمز لغوياً تعني شيء يمثل شيء أخر. مثلاً عندما تذهب للتقدم إلى وظيفة ما, يطلب منك صاحب العمل نسخة (صورة) من الشهادات التي حصلت عليها. فصورة الشهادة تمثل وترمز إلى الشهادة الأصلية. وهذه هى الفكرة ..! أي يقول لك البرنامج كل ما تريد أن تضيفه إلى الفيلم ضعه أولاً بالمكتبة Library ثم خذ نسخة, Instance منه وضعها أينما تريد في فيلم. وبذلك يستطيع البرنامج تعقب كل العناصر المكوّنة للقطة ويسمح لك بالتحكم بها, كما أن الحجم الكلي للفيلم سوف يقل إن أنت استخدمت هذا العنصر عدة مرات لأن هناك فقط أصل واحد والبرنامج يقوم بعرضه في المكان الذي تحدده, بالإضافة إلى أنه تم تخصيص لكل رمز خط زمني Timeline مستقل خاص به منفصل عن الخط الزمني الرئيسي للفيلم لمزيد من التحكم والمرونة.

وبما أن برنامج الفلاش يعشق التنظيم فقد تم تقسيم عناصر الفيلم إلى ثلاثة أنوع من الرموز:

- صورة ثابتة Graphic: وهى أى صورة ثابتة لن تقوم بعمل أي حركة في الفيلم وممكن أن تكون جزء من أى رمز أخر.
- زر Button: وهو للإضافة التفاعلية إلى الفيلم فهو يستجيب لحركات الفأرة و عملية النقر عليها أو على مفتاح من لوحة المفاتيح.
- مُتحركة Movie Clip: وهي أي صورة أو شكل متحرك مُبرمج مُسبقاً أو سوف يتم برمجته أو تعديل الخط زمني الخاص به لتحريكه داخل أي إطار في الفيلم الحالي, فهو بمثابة فيلم داخل فيلم.

ملحوظة مهمة ..!

المُتحركة أو الـ Movie Clip سوف نطلق عليه من الآن فصاعداً موفي كليب لكي تكون مميزة وسط الشرح.

كما ترى فقد أتاحت فكرة الرموز للمصممين القدرة على التحكم الكامل بعناصر الفيلم وبرمجة الرموز من نوع Button و Movie Clip من خلال تطبيق أكواد الأكشن إسكربت عليها, وبذلك تخطى برنامج الفلاش فكرة استخدامه لمجرد عرض تتابع معين من اللقطات إلى القدرة على التوقف عند إطار معين وعمل ما تشاء من حركة وتفاعلية. وهذه هى الميزة الأهم من وراء فكرة الرموز بالإضافة إلى الإقتصاد والتحكم بحجم الملف بقدر الإمكان.

وتستطيع أن تستنتج من كل ما سبق أن إدماج لغة الأكشن إسكربت في برنامج الفلاش جعلت الرمز هو وحدة البناء الجديدة الرئيسية في برنامج الفلاش بدلاً من الإطارات, ولذلك لعمل حركة لشكل أو لصورة معينة من خلال الأكشن إسكربت داخل أي إطار لابد من أن يكون نوع العنصر موفي كليب القابل للبرمجة. إلا إذا أردت أن تنشيء زر فيكون من نوع الزر Button.

إذاً يمكنك تطبيق أكواد أكشن إسكربت على الإطارت مباشرةً وعلى الرموز من نوع الموفي كليب والأزرار بالإضافة إلى الحقول النصية المتغيرة Dynamic Text Field و ملفات الصوت.

ولكي تتعامل مع أي رمز من خلال الأكشن إسكربت لابد وأن تعطيه اسم مميز لكي تستطيع أن تناديه به وتتحدث إليه من خلال لوحة الأكشن إسكربت. كما قلت سابقاً إن أي رمز تنشأه يضاف إلى المكتبة Library ومن ثم كل ما هو موجود على المسرح الـ Stage ما هو إلا نسخ Instances من هذا الرمز الموجود في المكتبة ولذلك كل رمز تريد أن تتعامل معه من خلال الأكشن إسكربت لابد وأن تعطيه اسم مميز خاص به Instance Name ولعمل ذلك تابع المثال التالي ..!

مثال:

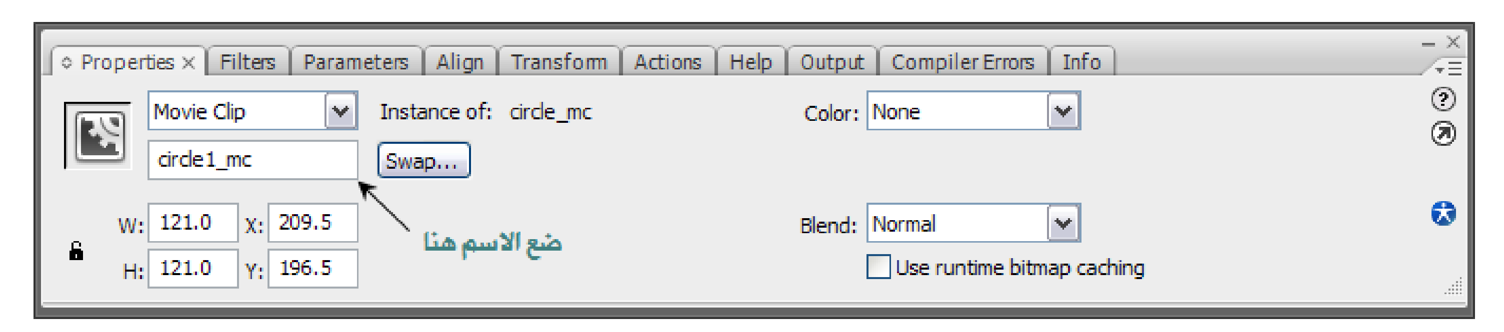
قم برسم دائرة مثلاً على المسرح ثم حولها إلى رمز من النوع موفي كليب ..؟!

- إما بالذهاب إلى قائمة Modify ومن تختار Convert to Symbol
- أو بالضغط على المفتاح F8

في كلتا الحالتين سيفتح لك البرنامج صندوق ادخل أي اسم تريده لهذا الرمز أن يظهر به في المكتبة وليكن circle_mc و حدد النوع Movie Clip وبهذا قد كوّنا النسخة الأصلية لهذا الرمز وإضيفت إلى المكتبة و تحولت الدائرة التي رسمنها على المسرح إلى مجرد نسخة من الأصل الموجود في المكتبة.

وللإعطاء اسم للنسخة الموجودة Instance Name أذهب إلى لوحة مستكشف الخصائص Properties Inspector, في الغالب ستجدها أسفل المسرح, وضع الاسم الذي تريد وليكن circle1_mc.

أنظر الصورة ..!



لاحظ أنه هناك عرف في التسمية حتى يستطيع المحرر مساعدتك أثناء كتابة الكود الخاص بهذا الرمز من خلال عرض تلميح عبارة عن قائمة بكافة الخصائص والدوال والأحداث الخاصة بهذا الرمز. فمثلاً في الاسم السابق إضافة المقطع _mc سوف يدرك المحرر سريعاً أنك تتعامل مع موفي كليب وسيعرض لك كل الخصائص والدوال والأحداث الخاصة بالرمز من النوع موفي كليب وهناك ايضاً مقاطع أخرى مثل:

_btn : للرمز من نوع زر.

_txt : للحقول النصية المتغيرة.

_sound : لملفات الصوت.

البرمجة الكائنية التوجه:

Object Oriented Programming...

لكي تستوعب مبدأ البرمجة الكائنية التوجه تابع معي المثال التالي أولاً ..!

لنفترض مثلاً أني قررت أن افتتح مركز لتعليم اللغات وقررت أن يقوم هذا المركز بتدريس ثلاث لغات (الإنجليزية والفرنسية والألمانية), لذلك قمت بتوظيف ثلاث فئات من المدرسين المتخصصين, الفئة الأولى عبارة عن مجموعة مدرسين لتدريس اللغة الإنجليزية والفئة الثانية عبارة عن مجموعة للفرنسية والفئة الأخيرة لتدريس الألمانية. مع أن كل المدرسين الذين يدرسون اللغة الإنجليزية تجمعهم صفة تدريس اللغة الإنجليزية في فئة واحدة إلا أنهم مختلفين فكل منهم له خصائص مختلفة كل منهم له اسمه الخاص وشكله الخاص وطابعه الخاص وحتى في مواعيد العمل في هذا المركز وكذلك الأمر مع فئة مدرسين اللغة الفرنسية والألمانية. ولكنهم يجمعهم شيء أخر وهي الآليات, فكل مدرس لغة إنجليزية مثلاً في هذا المركز يملك آلية (كيفية) القيام بالتدريس وكيفية تمرين الطلاب على القراءة والكتابة والإستماع. فكلها وظائف وآليات كل مدرس يتقنها جيداً ويعرف كيف يقوم بها. وكل مدرس ايضاً له القدرة على الإستجابة لأي سؤال يطرحه الطلاب عليه من نطاق تخصصه بطريقة شرح مناسبة.

إذاً مما سبق نستنتج أن لكل مدرس في فئة معينة لابد وأن له الآتي:

خصائص وآليات وأحداث يستجيب لها.

وهذه ببساطة هي الفكرة التي قامت عليها البرمجة الكائنية التوجه ..!

ولنطبق المثال السابق من منظور لغة الأكشن إسكريبت:

- تعتبر لغة الأكشن إسكريبت أن كل عنصر في اللقطة هو كائن Object, في مثالنا السابق أي مدرس في هذا المركز هو كائن (موظف) في هذا المركز.
- هذا الكائن ينتمي إلى فئة معينة Class, في مثالنا السابق قسمنا المدرسين إلى ثلاث فئات (فئة مدرسي اللغة الإنجليزية وفئة مدرسي الفرنسية وفئة مدرسي الألمانية).
- هذه الفئة هي التي تحدد وتصف كل الخصائص Properties والآليات Methods والأحداث Events التى تستجيب لها هذه الكائنات. يقدم الأكشن إسكريبت العديد من الفئات الجاهزة, ولكن تستطيع أنت ايضاً كتابة فئات جديدة خاصة بك.

مثال:

أي شكل ولنفترض مثلاً مربع من النوع موفي كليب هو كائن ينتمي إلى فئة الموفي كليب MovieClip ومع ذلك فله خصائص مختلفة عن أي موفي كليب آخر فمثلاً له كأي موفي كليب إحداثيات ولكن له إحداثي أفقي x وإحداثي راسي y (موضع) مختلف عن أي موفي كليب موجود في الفيلم وله أبعاد مثل أي موفي كليب ولكن له شكل مميز مختلف عن أي موفي كليب أخر وله خط زمني خاص به يحوي على عدد معين من الإطارت. كما أنك تستطيع التحكم في خطه الزمني من خلال آليات مبرمجه مسبقاً في فئة الـ MovieClip. ويمكنك أيضاً أن تجعله يستجيب لحركة الفارة او الضغط على مفتاح من لوحة المفاتيح أو إلى ما هنالك من أحداث.

ويمكن تلخيص ما سبق كالآتي:

- الخصائص Properties هى كل المتغيرات التي تميز الكائن كالأحداثيات, اللون ودرجة الشفافية إلى آخره.
- الأليات Methods هى كل الدوال المحددة لكل ما يستطيع الكائن أن يقوم به.
- الأحداث Events هى كل ما يستطيع الكائن الإستجابة له.

ملحوظة مهمة ..!

- فالحقيقة فكرة الكائنات تتعدى الموفي كليب والأزرار والحقول النصية المتغيرة. فالأكشن إسكريبت ينظر إلى أي لون كأنه كائن ولأي نوع خط كأنه كائن لأنهم في الأخير ينتموا إلى فئة الألوان وفئة الخطوط التى تحتوي على كافة الدوال التي تحدد كيفية ظهور هذا اللون أو ذاك الخط. أي أنه ليس بالضرورة أن يكون شيء يمكن رؤيته. فالأكشن إسكريبت مهوس بعمل تصنيف وفئات لكل شيء ممكن أن يخطر بالك أو لا يخطر, فهناك فئات لكل حدث من الأحداث وهناك فئات لوقت والتاريخ إلخ ..!

- الدالة, Function, هى مجموعة أوامر متسلسة ومرتبطة ببعضها تدل البرنامج على خطوات وكيفية تنفيذ هدف أو فكرة ما. فهى بمثابة تجميع أو تغليف لهذه الأكواد المرتبطة ببعضها لتكون ككتلة واحدة وأثناء كتابتك للدالة فإنك تعطي لها اسماً وكلما إحتجت لها فقط تستدعيها عن طريق كتابة اسمها فقط بدلاً من إعادة كتابة كل هذه الأوامر (الخطوات) مرة ثانية. والجدير بالذكر أن هناك الكثير من الدوال الجاهزة داخل اللغة التي نستدعيها عن طريق كتابة اسمها وهناك ما سوف ننشئه على حسب الرغبة أو السياق.

- المتغير, Variable, ماهو إلا عملية تسمية لأي قيمة من أجل تتبعها أو تغيرها فيما بعد. ومثال على ذلك الخصائص, فأثناء إعداد مبرمجي الأكشن إسكريبت لفئة الموفي كليب مثلاً قاموا بعمل عدة متغيرات (اسماء) لكي تشير إلى قيم معينة مثلاً متغيرين (اسمين) لقيم الإحداثيات, وأخر للون, وأخر لدرجة الشفافية إلى أخره.. وأنت مثلاً عندما تنشىء موفي كليب تستطيع بإستخدام هذه المتغيرات الإشارة (الاسماء) إلى واستعمال أو تغير قيم هذه الإحداثيات ومن ثم اسمها متغيرات لأنك تستطيع تتبع وتغير أي قيمة من خلالها. ولذلك فالمتغير مجرد اسم لأي شيء تريد التحكم به فيما بعد, ممكن أن يكون اسم لكائن معين وممكن أن يكون اسم لقيمة (خاصية) متعلقة بهذا الكائن ولذلك فللمتغيرات أنواع كثيرة.

سوف يأتي الحديث عن المتغيرات بالتفصيل فيما بعد ..! ولكن الجدير بالذكر هنا أنه يمكنك أن تنشىء ما تشاء من المتغيرات للإشارة إلى وتتبع كافة القيم التي تريد اثناء كتابتك للأكواد و عند استعمالك لهذه المتغيرات داخل الأكواد فإن الأكشن إسكريبت يقوم بإستبدال هذه الاسماء بالقيم التي تشير إليها وقت التنفيذ.

فكما ترى عشق الأكشن إسكريبت للتلاعب بالاسماء. فالدالة إن وجدت داخل الفئة أصبح اسمها آلية والمتغيرات أن وجدت داخل الفئة اصبح اسمها خصائص. وهناك مزيد من تحوّرات الاسماء سوف تجدها ولذلك لاتقلق فالأمر مجرد تغير اسم ..!

بعد كل هذا الكلام النظري فلنبدأ بالإنتقال إلى عالم الأكواد ..!

الخصائص:

...Properties

قم برسم مربع وضعه في منتصف المسرح ثم حوله إلى موفي كليب وأعطي النسخة اسم square_mc.

حدد الإطار رقم 1 في الخط الزمني, الـ Timeline, وأفتح لوحة محرر الأكشن إسكريبت وأكتب الكود التالي:

```
square_mc.y = 0;
trace(square_mc.x);
```

ولنبدأ في شرح السطرين البرمجيين السابقين:

- في السطر الأول قمنا بالنداء على الرمز الذي نريد التعامل معه بإسمه وهو square_mc. ولما حضر وضعنا نقطة. قصة هذه النقطة تحتاج إلى وقفة, هذه النقطة تسمى نقطة الإحتواء. أي أننا بوضعنا لهذه النقطة نقول للبرنامج أن الكلمة التالية سوف تكون مرتبطة وموجهة إلى هذا الرمز الذي قمنا بالنداء عليه. وهذه الكلمة أحدى ما تحويه الفئة التي ينتمي إليها هذا الرمز من خصائص أو دوال.
- ثم قمنا بكتابة y وهي تمثل الإحداثي الرأسي لهذا الرمز, فكما قلنا أن الإحداثيات هي أحد خصائص الرمز من النوع موفي كليب. وتلاحظ أنها أصبحت باللون الأزرق لأنها من الكلمات التي يتعرف عليها الأكشن إسكربت لأنها ببساطة تمثل شيء مبرمج مسبقاً في فئة الموفي كليب ولذلك فهى من الكلمات الرئيسية والمحجوزة للأكشن إسكربت ولا يمكنك استخدمها في تسمية أشياء جديدة.
- ثم جعلنها قيمة هذه الخاصية تساوي 0 ووضعنا الفاصلة المنقوطة ; ووجود هذه الفاصلة المنقوطة تعني للبرنامج أن الأمر البرمجي قد أنتهى. إذاً لتتعلم من الآن فصاعداً أن لا تنسى وضع الفاصلة المنقوطة عند نهاية كل أمر برمجي.

إذاً المعنى المقصود من السطر البرمجي الأول هو أننا نقول للبرنامج أجعل الإحداثي الرأسي الخاص بالرمز square_mc يساوي 0.

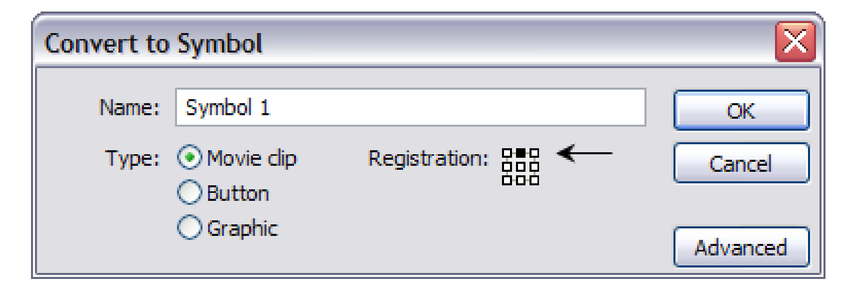
- في السطر الثاني استعملنا دالة التتبع ()trace وهي من أهم دوال الأكشن إسكربت التي تُستخدم في إختبار الكود أثناء البرمجة وهدفها الأساسي هو عرض معلومات عن قيمة مُتغير أو خاصية ما عند نقطة معينة. أي دالة في البرمجة تقبل شيء يسمى المعاملات تكتب بين قوسين وهذه المعاملات ماهي إلا متغيرات (اسماء) لقيم تمرر أو تعطى للدالة من الخارج وتحدد كيفية عمل هذه الدالة. دالة التتبع ()trace تقبل معامل يمثل ما نريده أن يًعرض في لوحة التتبع Output panel, في المثال السابق كان المعامل الذي وضعناه هو تعبير عن قيمة الإحداثي الأفقي الخاص بالرمز square_mc.

إذاً المعنى المقصود من السطر البرمجي الثاني هو أننا نقول للبرنامج أعرض في لوحة التتبع قيمة الإحداثي الأفقي الخاص بالرمز square_mc.

وعند إختبار الفيلم ستلاحظ ظهور لوحة التتبع Output panel وفيها قيمة الإحداثي الأفقي.
وأن الرمز قد أرتفع إلى أعلى, ولكنك تلاحظ اختفاء نصف المربع ..!

ولتفسير ذلك علينا ترجمة السطر البرمجي الأول بشكل دقيق ليكون كالآتي:
أجعل نقطة الربط Registration Point لهذا الرمز تساوي 0 على المحور الرأسي.

وبما أنك تركت نقطة الربط Registration Point في المنتصف (الوضع الإفتراضي) أثناء تحويلك للمربع إلى موفي كليب إذاً أختفي نصف المربع ..! ولتغيير ذلك عليك أن تجعل نقطة الربط في الأعلى أثناء تحويلك للمربع إلى موفي كليب. أنظر الصورة ..!



إذاً مما سبق قد تعلمت كيفية النداء على الرمز والتحدث إليه ..!

الآلـــــيات:

...Methods

كما ذكرت من قبل فإن الموفي كليب له خط زمني مستقل والذي يحتوي على عدد معين من الإطارت. هناك ما يسمى بمؤشر التشغيل Playhead يتحرك بالتسلسل بسرعة ثابتة على طول هذا الخط الزمني بالترتيب من أول إطار إلى آخر إطار ليعرض ما يحتويه كل إطار وما أن يصل إلى النهاية يعيد الكرَّة فيتحرك مرة أخرى من أول إطار إلى آخر إطار, أي أن مؤشر التشغيل مُعد ليكرر تشغيل الحركة الخاصة بالموفي كليب إلى ما لا نهاية. ولكن لحسن الحظ فهناك آليات ضمن فئة الموفي كليب لإتاحة فرصة للمبرمج للتحكم في كيفية تشغيل الخط الزمني للموفي كليب. فهناك دالة لتوقيف مؤشر التشغيل عند إطار معين وأيضاً هناك دالة لجعل المؤشر يبدأ التشغيل من عند إطار محدد وليس من الإطار الأول وغيرها الكثير ..!

1. دالة التوقف ()stop:

لنفترض مثلاً أن لديك موفي كليب اسمه circle_mc في خطه الزمني 50 إطار وهذا الموفي كليب له حركة عبارة عن الإنتقال من اليسار إلى اليمين وهذه الحركة تتكرر بإستمرار طبقاً للوضع الإفتراضي لمؤشر التشغيل, وتريد أن توقف هذه الحركة إذاً فسوف تضع الكود التالي في الخط الزمني الرئيسي للفيلم.

```
circle_mc.stop();
```

تماماً مثل القاعدة العامة قمنا بالنداء على الموفي كليب باسمه, ثم وضعنا نقطة الإحتواء للربط بين الدالة والموفي كليب, ثم قمنا بإستدعاء الدالة عن طريق كتابة اسمها. ولكنك تلاحظ أننا وضعنا قوسين فارغين .. لماذا ..؟!

هذه الأقواس الهلالية تسمى عموماً في لغات البرمجة أقواس الملحقات وهي جزء اساسي عند استدعاء الدالة. فكما قلت سابقاً أن أي دالة تقبل ما يسمى بالمعاملات ولكن في هذه الوضعية فإن هذه الدالة لن تحتاج إلى أي معاملات. أي أن المعاملات من ضمن ملحقات الدالة والمقصود أن هناك دوال تحتاج إلى معاملات نكتبها بين أقواس الملحقات وأخرى لا تحتاج فنترك أقواس الملحقات فارغة ..!

ثم قمنا بإنهاء الأمر بالفاصلة المنقوطة كما هو المعتاد.

ويمكن أن تتسأل لماذا نكتب قيم هذه المعاملات (المتغيرات) داخل الأقواس بهذه الطريقة ..؟!

الجواب ببساطة هو لضمان أن المُبرمج قام بكتابة قيمة لهذا المعامل ولم ينساها لأني كما قلت ان عمل الدلة قد يتوقف على وجود هذا المعامل.

2. دالة الإنتقال والتشغيل ()gotoAndPlay:

في المثال السابق مثلاً لو اردنا أن نجعل مؤشر التشغيل ينتقل إلى الإطار رقم 25 ويبدا التشغيل من هناك بدلاً من البدء من الإطار الأول فسوف نستخدم دالة الإنتقال والتشغيل:

```
circle_mc.gotoAndPlay(25);
```

السياق البرمجي للأمر السابق كما هو المعتاد ولكنك تلاحظ أن الدالة تقبل معامل وهو يمثل رقم الإطار الذي يجب أن تنقل إليه الدالة مؤشر التشغيل ليبدأ التشغيل من عنده.

- ولكني أيضا أريدك أن تلاحظ أن اسم الدالة السابقة رغم أنها عبارة إلى 3 مقاطع إلا أننا لم نكتب مثلاً goto and play ففي لغة الأكشن إسكربت هناك عُرف في التسمية, فبالنسبة للدوال والمتغيرات فإنه لا توجد مسافات والمقاطع تكون ملاصقة لبعضها ويبدأ المقطع الأول بحرف صغير ومن ثم كل مقطع جديد يبدأ بحرف كبير. واسم هذه الدالة نموذج لهذا العُرف المُتبع. وإن قمت بإستدعاء اسم دالة مثلاً ولم ترعي الحروف الكبيرة والصغيرة فسيكون هناك خطأ ولن يتعرف الأكشن إسكربت على الدالة. لأن gotoAndPlay خلاف gotoandplay حيث أن لغة الأكشن إسكربت حساسة لحالة الحرف. ويمكنك أن تلاحظ ذلك إذا لم يتغير لون اسم الدالة إلى اللون الأزرق.

- أما بالنسبة لعُرف تسمية الفئات في لغة الأكشن إسكربت فأيضاً لا توجد مسافات والمقاطع تكون ملاصقة لبعضها إلا انه كل مقطع يبدأ بحرف كبير فمثلاً فئة الموفي كليب اسمها MovieClip, فكما تلاحظ المقطع الأول يبدأ بحرف كبير والثاني ايضاً و هكذا ...

- هناك ايضاً ما يسمى بالثوابت وهذه الثوابت ماهي إلا (اسماء) خصائص تماماً مثل المتغيرات إلا أن القيم التي تشير إليها ثابتة لا يمكن تغيرها, على العكس من الخصائص التي رأيتها في الأعلى كالإحداثيات, واسماء هذه الثوابت تكتب كلها بحروف كبيرة وإن كانت عبارة عن عدة مقاطع فإنها تفصل بشرطة تحتية مثلاً ENTER_FRAME.

لاتقلق فسوف يتضح لك الأمر بعد قليل ..! المهم أن تعرف كيف تُميز بين الدوال والمتغيرات والفئات والثوابت.

3. دالة الإنتقال والتوقف ()gotoAndStop:

في المثال السابق ايضاً لو اردنا أن نجعل مؤشر التشغيل ينتقل إلى الإطار رقم 25 ولكن في هذه المرة يقف هناك بدلاً من الإستمرار في التحرك فسوف نستخدم دالة الإنتقال والتوقف:

```
circle_mc.gotoAndStop(25);
```

وأعتقد أنه إلى الآن الكود السابق واضح جداً ..!

الأحـــداث:

...Events

قلنا فيما سبق أن الأحداث هي كل ما يستطيع الكائن الإستجابة له. يستجيب الكائن إما للأحداث من قبل المستخدم, مثل الضغط على مفتاح من لوحة المفاتيح أو النقر على زر الفأرة, أو أحداث داخل الفيلم, والتي تعتمد على الخط الزمني أو عامل وقتي مثل المؤقتات Timers.

لكي يتضح لك كيفية تعامل الأكشن إسكربت مع الأحداث تابع المثال التالي:

مثلاً في مركز الإطفاء , عندما يتم قرع الجرس يهّب رجال الإطفاء ويسرعوا إلى سيارة الإطفاء ويتوجهوا إلى مكان حدوث الحريق. فلماذا فعلوا ذلك ..؟!! لآنهم ببساطة قد دُربوا على إنتظار قرع الجرس الذي يكون بمثابة الإبلاغ عن وقوع حادثاً في مكاناً ما. وهذه هي الفكرة ببساطة...!

ولنطبق هذه الفكرة من خلال الأكشن إسكربت:

هناك ما يسمي بدالة إنتظار حدث Event Listener وهي ببساطة دالة يتم إدراجها في قائمة إنتظار لتُنفذ عند وقوع حدث ما. في المثال السابق الدالة هي الإسراع إلى سيارة الإطفاء والتوجه إلى مكان حدوث الحريق أما الحدث فهو عند قرع الجرس. فعند وقوع هذا الحدث فإنه يتم إبلاغ الأكشن إسكربت بوقوعه ليقوم بتفيذ كل دوال الإنتظار المدرجة في قائمة إنتظار وقوع هذا الحدث لينفذها ولذلك تُسمى ايضاً بدالة التوجيه Handler لأنها ستوجه البرنامج لخطوات عمل شيء معين عند وقوع هذا الحدث (مثلاً عند النقر على زر الفآرة).

أما الخطوات الفعلية التي يتبعها الأكشن إسكربت فهي كالتالي:

1. **إدراج دالة إنتظار حدث Event Listener في قائمة إنتظار وقوع حدث ما, لتُنفذ عند وقوع هذا الحدث.**
2. **عـند وقوع الحدث المنتظر, فإنه يتم إنشاء كائن حدث Event Object جديد من فئة هذا الحدث Event Class. هذا الكائن بمثابة الشاهد الذي يعرف كل ملابسات هذا الحدث وكيف تم. فمثلاً بالنسبة (لحدث متعلق بالفأرة) فإن كائن الحدث, الذي ينتمي إلى فئة أحداث الفأرة MouseEvent Class, سوف يحوي عدة متغيرات مثل التي تحدد إحداثيات (مكان) مؤشر الفأرة وقت وقوع هذا الحدث إلخ ...!**
3. **بعد ذلك تتم عملية الإبلاغ التي تُسمى عملية بث (خبر وقوع) الحدث Event Dispatch إلى كل دوال الإنتظار وهي بمثابة الضوء الأخضر للأكشن إسكربت ليبدأ تنفيذ دوال الإنتظار.**
4. **أثناء عملية بث الحدث فإن كل دوال الإنتظار المدرجة على قائمة الإنتظار يُمرر إليها كائن الحدث على هيئة معامل (متغير) ويتم بعد ذلك تنفيذها. وعملية تمرير هذا الكائن للدالة تمت لأنه ربما تحتاج الدالة إلى إستخدام متغيرات كائن الحدث للوصول إلى معلومات حول هذا الحدث.**

ملحوظة مهمة ..!

- **هناك ما يُسمى بمستقبل الحدث Event Target وهو ببساطة عبارة عن اي كائن سوف يستقبل الحدث. فمثلاً عند النقر على رمز من الرموز مثل الموفي كليب, في هذه الحالة عملية النقر تمثل الحدث والموفي كليب يمثل مستقبل الحدث. لكل مستقبل حدث قائمة إنتظار خاصة به تستطيع ببساطة أن تدرج ما تشاء من دوال إنتظار في هذه القائمة.**
- **كما تري في الخطوات الفعلية التي يتبعها الأكشن إسكربت في التعامل مع الأحداث فإني استخدمت أفعال كلها مبنيه للمجهول وكأنه هناك يد خفية هى التي تقوم بهذه العملية. في الواقع الإجابة نعم. فكما تري أننا قد ربطنا مُستقبل الحدث وهو عبارة عن كائن بدالة إنتظار حدث وهي عبارة عن مجرد دالة وهذه العملية لابد وأنها تمت من خلال وسيط ألا وهو فئة موجودة داخل الأكشن إسكربت تسمى فئة البث EventDispatcher Class. وهي التي تقوم بكل خطوات ربط وإنشاء علاقة تربط كائنين ببعضهما حيث أنها توفر الآليات المناسبة لعمل تلك الخطوات من الإدراج وإلغاء الإدراج. ولذلك فإن الأكشن إسكربت يقوم بتوريث الكائن المُستقبل للحدث كل أليات فئة EventDispatcher Class لكي يتمكن من التفاعل مع دالة الإنتظار.**

ويمكنك تخليص الفكرة السابقة كالآتي:

اولاً نقوم بإدارج الدالة التي نريد في قائمة إنتظار خاصة بكائن معين ومن ثم عند حدوث الحدث يتم إبلاغ الأكشن إسكربت ليبدأ تنفيذ الدالة.

1.الأحداث الخاصة بالفأرة:

قم برسم مربع وضعه في منتصف المسرح ثم حوله إلى موفي كليب وأعطي النسخة اسم square_mc.

حدد الإطار رقم 1 في الخط الزمني, الـ Timeline, وأفتح لوحة محرر الأكشن إسكربت وأكتب الكود التالي:

```
square_mc.addEventListener(MouseEvent.CLICK , onClick);
function onClick (evt:MouseEvent):void {
    trace (evt.type);
}
```

ولنبدأ في شرح الكود السابق:

- في السطر البرمجي الأول قمنا بالنداء على الرمز الذي سيستقبل الحدث باسمه وهو square_mc. ولما حضر وضعنا نقطة الإحتواء.
- وكما قلت سابقاً أن الكائن الذي يستقبل الحدث يرث كل الآليات الموجودة بفئة البث EventDispatcher Class المسئولة عن إدارة الحدث. ولذلك استدعينا دالة ()addEvenListener بعد النقطة والتي أصبحت جزء من آليات هذا الرمز بعد عملية التوريث من فئة البث. وهذه الدالة هى المسئولة عن إدارج دالة إنتظار حدث في قائمة الإنتظار الخاصة بهذا الرمز.
- هذه الدالة تقبل معاملين, المعامل الأول هو اسم الحدث الذي سيستقبله هذا الرمز, والمعامل الأخر هو اسم الدالة التي نريد إدراجها في قائمة إنتظار وقوع هذا الحدث على هذا الرمز ليتم تنفيذها.
- كل حدث ينتمي إلى فئة معينة, فأحداث الفأرة تنتمي إلى فئة MouseEvent وهذه الفئة تُضم أنواع كثيرة من الأحداث ولكل حدث اسم بالطبع. وبما أن اسم الحدث ثابت لا يمكن أن تغيره فقد وضعه مطوري لغة الأكشن إسكربت فيما يسمى بالثابت Constant. ولذلك لكي نصل إلى اسم الحدث مثل حدث النقر على الفأرة فالثابت الخاص بهذا الحدث هو CLICK ولذلك نستخدم التعبير MouseEvent.CLICK. وعند التنفيذ يقوم الأكشن إسكربت بإستبدال التعبير MouseEvent.CLICK بقيمته وهى اسم الحدث click.
- كل ما قمنا به في الكود السابق هو أننا كتبنا بين أقواس الملحقات اسم الفئة التي ينتمي إليها الحدث وهي MouseEvent ثم وضعنا نقطة الإحتواء وحددنا اسم الثابت (الخاصية) الذي يمثل اسم الحدث الذي نريده وهو CLICK أى عند النقر مرة واحدة على الزر الأيسر للفأرة. ثم وضعنا فاصلة لنكتب المعامل الثاني وهو اسم دالة الإنتظار في هذه الحالة سميتها أنا onClick ولك الحرية في إختيار الاسم الذي تريد.
- بالطبع يمكن بدلاً من كتابة MouseEvent.CLICK أن تكتب "click" مباشرةً وسيكون لونها أخضر لآنها بين علامتي تنصيص وهذا يعني أنها مجرد قيمة (اسم) نصية ولكن الصيغة الأولى أدق وأفضل حتى يتمكن الأكشن إسكربت من مساعدتك في حالة كتابة اسم الحدث خطأ. لأن الصيغة الثانية سوف يقبلها كما هى حتى لو كانت خطأ ولن ينبهك ولن تستطيع تتبع الخطأ أما عند الخطأ في كتابة الصيغة الأولى سوف ينبهك الأكشن إسكربت لمكان الخطأ ويحدده لك.

إذاً المعنى المقصود من السطر البرمجي الأول هو أننا نقول للبرنامج قم بإدراج الدالة onClick في قائمة إنتظار وقوع حدث النقر على الرمز square_mc.

- في السطر البرمجي الثاني قمنا بإنشاء الدالة التي تُسمى onClick. في بداية الأمر لكي نُعرف دالة جديدة كتبنا كلمة function ثم قمنا بتحديد اسمها وهو onClick.
- كما ذكرت فإنه بمجرد وقوع الحدث يتم عمل كائن حدث الذي يُمرر إلى الدالة ولذلك لابد من إعطاءه اسماً لكي يتم التعامل معه. وكما تعلمت في السابق فإن كلمة اسم تساوي متغير. إذاً انشائنا متغير بين اقواس الملحقات سميته أنا evt ولك الحرية في اختيار ما تشاء. ثم وضعنا نقطتين, في لغات البرمجة عموماً هاتين النقطتين تعنى أننا سوف نحدد نوع شيء ما, ولذلك حددنا أن evt سيكون اسماً لكائن حدث من فئة MouseEvent. إذاً حددنا للأكشن إسكربت اسماً للكائن الذي يمرره لهذه الدالة.
- ثم وضعنا نقطتين, تماماً مثلما يدور في ذهنك الآن, فهي لتحديد نوع هذه الدالة. في هذه الحالة فإن الدالة ستكون دالة خالية Void أي أن الدالة ستكون خالية من نتيجة نهائية. فهى فقط ستقوم بتنفيذ تتابع معين من أوامر وليست هناك قيمة نهائية ترجعها الدالة. فقد يكون مثلاً لديك دالة تحسب تاريخ معين ثم ترجع لك هذا التاريخ كقيمة نهائية لحسابات معينة وتستطيع ان تستخدم هذا التاريخ فيما بعد وفي هذه الحالة تستبدل كلمة Void بنوع القيمة التي ترجعها الدالة إذا كانت نص فتكون String وإذا كانت رقم فتكون Number.
- ثم وضعنا قوس معكوف, هذه الأقواس عموماً في لغات البرمجة تسمى أقواس التغليف, وهى تستخدم لتغليف مجموعة أكواد. اذاً لبدأ كتابة أكواد هذه الدالة فتحنا أقوس التغليف.

إذاً المعنى المقصود من السطر البرمجي الثاني هو أننا نقول للبرنامج قم بإنشاء دالة اسمها onClick وسيكون لها معامل عبارة عن اسم للكائن من الفئة MouseEvent الذي سوف يمرر إليها وهذه الدالة ستكون دالة خالية.

- في السطر البرمجي الثالث استعملنا دالة ()trace وجعلنا المعامل هو evt.type وهو التعبير الذي يُمثل (نوع) الحدث. فكما ترى فقد استعملنا اسم كائن الحدث ثم وضعنا نقطة الإحتواء وكتبنا type وهو ثابت ضمن كائن الحدث يمثل نوع الحدث.

إذاً المعنى المقصود من السطر البرمجي الثالث هو أننا نقول للبرنامج أعرض في لوحة التتبع قيمة الثابت type الخاص بهذا الحدث.

- ثم قفلنا أقواس التغليف.

وعند إختبار الفيلم والنقر على المربع ستلاحظ ظهور لوحة التتبع Output panel وفيها كلمة Click وهي قيمة الثابت type الخاص بهذا الحدث.

كما قلت فهناك أنواع كثير للأحداث داخل فئة MouseEvent ذكرنا منها CLICK اما بالنسبة لباقي الأحداث فالثوابت الخاص بهم هي:

DOUBLE_CLICK
MOUSE_DOWN
MOUSE_MOVE
MOUSE_OUT
MOUSE_OVER
MOUSE_UP
MOUSE_WHEEL
ROLL_OUT
ROLL_OVER

2.الأحداث الخاصة بلوحة المفاتيح:

قم برسم مربع وضعه في منتصف المسرح ثم حوله إلى موفي كليب وأعطي النسخة اسم square_mc.

حدد الإطار رقم 1 في الخط الزمني, الـ Timeline, وأفتح لوحة محرر الأكشن إسكربت وأكتب الكود التالي:

```
stage.addEventListener(KeyboardEvent.KEY_DOWN , onClick);
function onClick (evt:KeyboardEvent):void {
    square_mc.x = 0 ;
}
```

بالنسبة لشرح الكود السابق:

- الفكرة تماماً مثل أحداث الفأرة إلا أننا جعلنا مستقبل الحدث بدل من المربع square_mc جعلناه المسرح الـ stage, لأنه لكي يستجيب المربع لحدث الضغط على مفتاح من لوحة المفاتيح لابد من النقر على المربع أولاً لتحديده. وهذا امر بالطبع غير منطقي ولذلك جعلنا مستقبل الحدث هو المسرح.
- وبالنسبة للحدث فهو ضمن فئة KeyboardEvent والثابت الذي يمثل اسم حدث الضغط على لوحة المفاتيح هو KEY_DOWN.
- المختلف فقط عن الكود السابق هو اننا جعلنا دالة onClick تقوم بجعل الأحداثي الأفقي للمربع يساوي 0.

وعند إختبار الفيلم والضغط على المسطرة مثلاً من لوحة المفاتيح ستلاحظ إنتقال المربع إلى اليسار.

الجدير بالذكر هنا أنه يمكنك معرفة أي مفتاح قام المستخدم بالضغط عليه. فكما قلت أن فائدة كائن الحدث الذي يُمرر إلى دالة الإنتظار هو توفير معلومات للدالة عن الحدث الذي تم من خلال عدة خصائص. ولذلك فهناك خاصيتين, الأولى وهي charCode وهي تمثل المفتاح الذي قام المستخدم بالضغط عليه فمثلاً عند الضغط على مفتاح الحرف a فإن قيمة هذه الخاصية تصبح "a" . والخاصية الثانية هي keyCode وهي تعطي رقم المفتاح الفعلي على لوحة المفاتيح فمثلاً عند الضغط على مفتاح الحرف a فإن قيمة هذه الخاصية تصبح 65.

3.الحركة بإستخدام الحدث ENTER_FRAME:

قد تحدثنا عن الأحداث المرتبطة بالمستخدم مثل أحداث الفأرة وأحداث لوحة المفاتيح وكيفية عملها. أما النوع الثالث من الأحداث فهو أحداث داخل الفيلم, والتي تعتمد على الخط الزمني. الفكرة بمنتهى البساطة قائمة على جعل البرنامج ينتظر حدوث شيء جديد ولكي تستوعب هذه الفكرة لابد وأن تفهم كيفية عمل برنامج الفلاش في عرض الإطارات.

قد ذكرت أن الشق التصميمي قائم على عمل عدة لقطات (إطارات) مختلفة ومتتابعة وكل إطار يحتوي على عناصر معينة مكوًنة له ويقوم رأس التشغيل بتحرك لعرض محتويات تلك الإطارات بسرعة معينة. فلو فرضنا مثلآ اننا لدينا فيلم مكوًن من 5 إطارت وسرعة عرض تلك الإطارات هي (إطار واحد كل ثانية), إذاً مدة عرض الفيلم ستكون 5 ثواني. ولكن كيف يقوم برنامج الفلاش بعمل ذلك. فالحقيقة يقوم بشيء شبيه بالتالي عند تشغيل هذا الفيلم:

1. في الثانية الأولى يعرض محتويات الإطار الأول.
2. وعند الثانية الثانية يفحص العناصر في الإطار الثاني فإن وجد تغيير في أحد عناصر الإطار الثاني فيقوم بتحديث الشاشة وعرض التغيير وإن كان الإطار الثاني مشابه تماماً للإطار الأول فيبقى الحال كما هو عليه ولا يعرض الإطار الثاني بل ينتظر للثانية الثالثة لفحص محتويات الإطار الثالث.
3. وعند الثانية الثالثة يبدأ الفحص فإن وجد تغيير في أحد عناصر الإطار الثالث يقوم بتحديث الشاشة ويعرض العناصر التي تغيرت فقط في الإطار الثالث والا فإنه يبقى على الوضع الحالي وينتظر للثانية الرابعة وهكذا حتى نهاية الفيلم ...!

ولكني قد ذكرت ايضاً أنه يمكنك إضافة أكواد أكشن إسكربت إلى أى إطار, ولأنه من الممكن أن تقوم هذه الأكواد بعمل تغير في أحد العناصر المكوّنة للإطار فإن أكواد الإكشن إسكربت لكل إطار تُنفذ أولاً من أول سطر إلى أخر سطر قبل عمل فحص وتحديث الشاشة (عرض النتيجة النهائية). والجدير بالذكر أن عملية تنفيذ أكواد الأكشن إسكربت تستغرق وقتاً ففي المثال السابق مثلاً لنتفرض أن برنامج الفلاش استغرق وقتاً قدره 100 ميللي ثانية لتنفيذ أكواد الإطار الأول وهذه الأكواد قامت بعمل تغير في أحد عناصر هذا الإطار وكما قلت أننا في المثال السابق قمنا بعمل سرعة عرض الإطارت (إطار واحد كل ثانية). لذلك فإن برنامج الفلاش سوف ينتظر 900 ميللي ثانية (مرور أول ثانية) قبل أن يبدا بعرض محتويات الإطار الأول. وحتي عملية العرض هذه تستغرق وقتاً ولنفترض مثلاً أنه قد استغرقت 50 ميللي ثانية لكي يكمل عرض الإطار الأول. إذاً استغرقت عملية إعداد وعرض الإطار الأول 1050 ميللى ثانية منذ بدأ تشغيل الفيلم. وهكذا مع باقي الإطارات ...!

أي ان المقصود من الكلام السابق هو أن برنامج الفلاش لا يقوم بتنفيذ كل سطر برمجي على حدى ويعرض مباشرة التغيير الذي أحدثه هذا الأمر على الشاشة ثم يتنقل إلى السطر التالي, بل أن برنامج الفلاش ينفذ أوامر الأكشن إسكربت كلها دفعة واحدة ثم ينتظر حتى يحين الموعد المجدول لعرض الإطار ليقوم بعرض النتيجة النهاية للمحتويات الإطار والتغيير الذي أحدثته أكواد الأكشن إسكربت. ففي المثال السابق رغم أن برنامج الفلاش أتم تنفيذ كل أكواد الأكشن إسكربت الخاصة بالإطار في 100 ميللي ثانية إلا أنه انتظر 900 ميللي ثانية ليعرض النتيجة في الموعد المحدد للإنتقال إلى الإطار التالي (بعد مرور ثانية).

ومثال أخر لكي تتضح الصورة لنتفرض أن هناك دالة إنتظار حدث تُحرك موفي كليب عند النقر عليه إلى أقصى اليمين. وعند التشغيل مباشرة أتم الأكشن اسكربت تنفيذ كل أكواد الإطار الأول في 100 ميللي ثانية وأصبح مهيىء للإستقبال هذا الحدث. ولنتفرض أن المستخدم قام بالنقر على الموفي كليب عند 500 ميللي ثانية, فيقوم البرنامج بتنفيذ أوامر دالة الإنتظار ورغم ذلك فإن برنامج الفلاش لن يبدأ بتحريك الموفي كليب إلا عند 1000 ميللي ثانية.

أعتقد أصبح الأمر واضح الأن..!

ومن ثم فإن الوصف الدقيق لكيفية العرض والتنقل من إطار إلى أخر منذ بداية تشغيل الفيلم ستكون كالآتي :

1. يقوم برنامج الفلاش بتنفيذ كل أوامر الأكشن إسكربت في الإطار الأول.
2. ينتظر حتى يحين الموعد المجدول لعرض الإطار, وأثناء الإنتظار إذا وقع حدث مبرمج مسبقاً يقوم بتنفيذ كل دوال الإنتظار الخاصة بهذا الحدث.
3. حينما يحين موعد عرض الإطار يقوم بفحص إذا كانت الشاشة تحتاج إلى تحديث. وتحتاج الشاشة إلى تحديث في الظروف التالية:
    1. إذا كان هناك تغيير في عناصر الإطار قد تم من خلال الشق التصميمي.
    2. إذا قامت أكواد الأكشن إسكربت الخاصة بالإطار بإحداث تغيير ما أو تعديل في الوضع الحالي لعناصر الإطار.
    3. إذا قامت دوال إنتظار حدث بإحداث تغيير ما أو تعديل في الوضع الحالي لعناصر الإطار.
4. بعد عملية الفحص السابقة إذا كانت هناك ضرورة لتحديث الشاشة بعناصر جديدة أو معدلة يقوم بتحديثها.
5. تكرار الخطوات من 1-4 مع الإطار الثاني وباقي الإطارات.

والجدير بالذكر أنه قبل كل عملية الفحص عن اشياء تستدعي تحديثاً للشاشة يقوم برنامج الفلاش ببث (خبر وقوع) حدث اسمه حدث الإنتقال إلى إطار جديد ENTER_FRAME ويعمل هذا الحدث كمنبه لكل العناصر التي تحتاج إلى تحديث وتنتظر لكي يتم عرضها. أى انه يقول قد حان الآن وقت العرض..!

استعرضت إلى الآن طريقة الفلاش في التعامل مع أكثر من إطار اثناء العرض, ولكنك في الأكشن إسكربت سيكون أغلب تعاملك مع إطار واحد فقط سواء أكان كل الفيلم عبارة عن إطار واحد أو أنك أوقفت الفيلم عند إطار معين. ولذلك لابد أن تتعرف على طريقة الفلاش في عرض فيلم عبارة عن إطار واحد فقط.

الخطوات التي يتبعها الفلاش في عرض إطار واحد مشابهة لحد كبير الخطوات السابقة ولكن مع بعض الإختلافات كما يلي:

1. يقوم برنامج الفلاش بتنفيذ كل أوامر الأكشن إسكربت في الإطار الأول.
2. ينتظر حتى يحين الموعد المجدول لعرض الإطار, وأثناء الإنتظار إذا وقع حدث مبرمج مسبقاً يقوم بتنفيذ كل دوال الإنتظار الخاصة بهذا الحدث.
3. حينما يحين موعد عرض الإطار يقوم بفحص إذا كانت الشاشة تحتاج إلى تحديث. وتحتاج الشاشة إلى تحديث في الظروف التالية:
    1. إذا كان هناك تغيير في عناصر الإطار تم من خلال الشق التصميمي.
    2. إذا قامت أكواد الأكشن إسكربت الخاصة بالإطار بإحداث تغيير ما أو تعديل في الوضع الحالي لعناصر الإطار.
    3. إذا قامت دوال إنتظار حدث بإحداث تغيير ما أو تعديل في الوضع الحالي لعناصر الإطار.
4. بعد عملية الفحص السابقة إذا كانت هناك ضرورة لتحديث الشاشة بعناصر جديدة أو معدلة يقوم بتحديثها.
5. ينتظر حتى يحين الموعد المجدول لعرض الإطار, وأثناء الإنتظار إذا وقع حدث مبرمج مسبقاً يقوم بتنفيذ كل دوال الإنتظار الخاصة بهذا الحدث.
6. حينما يحين موعد عرض الإطار يقوم بفحص إذا كانت الشاشة تحتاج إلى تحديث. وتحتاج الشاشة إلى تحديث كما ذكرنا فيما سبق
7. بعد عملية الفحص السابقة إذا كانت هناك ضرورة لتحديث الشاشة بعناصر جديدة أو معدلة يقوم بتحديثها.
8. ينتظر حتى يحين الموعد المجدول ...

كما ترى فإنه فقط ينتظر حدوث شيء جديد سواء أكان من خلال أكواد الأكشن إسكربت أو من خلال حدث تم في فترة الإنتظار. وبذلك يمكننا إستغلال الحدث ENTER_FRAME الذي تحدثنا عنه, بحيث قبل أن يقوم الفلاش بعملية الفحص عن تغير في الإطار ليقوم بعرضه نقوم نحن بإستجابة لحدث ENTER_FRAME ومن ثم نجعل الأكشن إسكربت ينفذ دالة انتظار تُحدث التغيير الذي نريده ليتم عرضه.

ولتوضيح الفكرة أكثر يمكنك أن تعتبر الحدث ENTER_FRAME كأنه حلقة تكرار تقوم بتحديث الشاشة بتغييرات معينة عدة مرات في الثانية. وعدد هذه المرات يساوي سرعة عرض الإطارات التي إعدادنها للفيلم.

مثال:

تخيل مثلاً لو أردنا أن نجعل موفي كليب يتحرك من اليسار إلى اليمين خطوة خطوة. إذاً فخيارنا الأكيد هو دالة إنتظار حدث ENTER_FRAME.
قم برسم مربع وضعه على يسار المسرح ثم حوله إلى موفي كليب وأعطي النسخة اسم square_mc.
حدد الإطار رقم 1 في الخط الزمني, الـ Timeline, وأفتح لوحة محرر الأكشن إسكربت وأكتب الكود التالي:

```
stage.addEventListener(Event.ENTER_FRAME ,onStart);
function onStart(evt:Event):void {
    square_mc.x += 1;
}
```

بالنسبة لشرح الكود السابق:

- الفكرة تماماً مثل الفكرة العامة للأحداث التي تكلمنا عنها وجعلنا مستقبل الحدث هو المسرح الـ stage.
- وبالنسبة للحدث فهو ضمن فئة Event التي تحوي على الأحداث العامة والثابت الذي يمثل اسم حدث الإنتقال إلى إطار جديد هو ENTER_FRAME. ودالة الإنتظار أسميناها onStart.
- قمنا بإنشاء الدالة و خصننا أسم evt لكائن الحدث من فئة Event.
- ولكي نجعل الأكشن إسكربت يزيد قيمة الإحداثي الرأسي للموفي كليب بمقدار واحد كل مرة استعملنا رمز الإضافة وهو += ثم وضعنا الرقم 1.

وعند إختبار الفيلم ستلاحظ تحرك المربع من اليسار إلى اليمن خطوة بخطوة. يمكنك أن تجعله يتحرك أسرع عن طريق زيادة سرعة عرض الإطارات مثلاً من 12 إلى 24 إطار في الثانية وهكذا..!
ولكنك ستلاحظ ايضاً بعد فترة إختفاء الموفي كليب لأنه استمر في التحرك وتخطى طول المحور الأفقي للمسرح. ففي الغالب الأعم يكون حجم الإفتراضي للمسرح هو 550 x 400. ولذلك علينا أن نجد طريقة لجعله يتوقف عن التحرك عندما يصل إلى 550 ..!
ويمكننا عمل ذلك عن طريق جمل الشرط If. وهي تستخدم لوضع شرط يتم التحقق منه قبل تنفيذ أمر ما.
ولذلك علينا تحديث الكود السابق ليصبح كالآتي.

```
stage.addEventListener(Event.ENTER_FRAME ,onStart);
function onStart(evt:Event):void {
    if (square_mc.x < 500)  {
        square_mc.x += 1;
    }
    else{
        stage.removeEventListener(Event.ENTER_FRAME ,onStart);
    }
}
```

بالنسبة لشرح الكود السابق:

- تلاحظ بالتأكيد أن الإختلاف موجود في أوامر دالة الإنتظار. بالتحديد أننا قمنا بإضافة جملة شرط.
- قمنا بعد إنشاء دالة الإنتظار بعمل جملة الشرط عن طريق كتابة الصيغة العامة وهي كلمة if وتلاحظ تحولها إلى اللون الأزرق. ثم في أقواس الملحقات وضعنا الشرط الذي نريده وهو كون الإحداثي الأفقي للموفي كليب أقل من 500. ثم وضعنا ما سوف يفعله الأكشن إسكربت إذا ما تحقق هذا الشرط بين أقواس التغليف وهو زيادة الإحداثي الأفقي بواحد.
- ثم كتبنا كلمة else وتعني إذا لم يتحقق الشرط ووضعنا ما سوف يفعله الأكشن إسكربت إذا لم يتحقق هذا الشرط بين أقواس التغليف.
- تلاحظ كيف جلعنا الأكشن إسكربت يتوقف عن تحريك الموفي كليب وهو عن طريق أننا استعملنا الدالة removeEventListener للإزالة دالة الإنتظار onStart من قائمة إنتظار الحدث ENTER_FRAME بحيث لا يتم تنفيذها إذا لم يتحقق الشرط الذي وضعناه.
- وطريقة كتابة الدالة removeEventListener تماماً مثل الدالة addEventListener وهي كتابة اسم مستقبل الحدث ثم الدالة removeEventListener التي سوف تزيل دالة الإنتظار من قائمة إنتظار الحدث والتي تم تحديدهما بين أقواس ملحقات الدالة.
- ثم أغلقنا أقواس التغليف الخاصة بدالة الإنتظار.

إذاً المعنى المقصود هو إذا تحقق الشرط وهو كوّن الإحداثي الأفقي للموفي كليب أقل من 500 إذاً قم زيادة الإحداثي الأفقي و**إلا** فقم بالتوقف عن الإستجابة لهذا الحدث.

وعند إختبار الفيلم ستلاحظ تحرك المربع من اليسار إلى اليمن خطوة بخطوة إلى أن يصل إلى نهاية المسرح فيتوقف.

أها ..! إلى الآن فقد قطعت شوطاً كبيراً في تعلم اساسيات الأكشن إسكربت 3.0 ولكن يتبقى بعد المتفرقات التي يجب أن تكون مُلم بها.

متفرقات:

لابد منها بالتأكيد ..

1.إنشاء كائن جديد من فئة:

تحدثت مرات عديدة عن هوس الأكشن إسكربت في عمل فئات لكل شيء ولذلك أنت إيضاً عليك التأقلم مع هذا الوضع. وتعاملك مع هذا المفهوم سيكون من خلال إنشاء كائنات جديدة من الفئات التي يوفرها لك الأكشن إسكربت.

مثال:

هناك فئة خاصة بالتاريخ والوقت تُسمى Date وللإنشاء كائن جديد يحتوي على كافة المعلومات عن التاريخ والوقت الحالي نقوم بإستعمال كلمة **new** ثم إسم الفئة التي نكوّن منها كائن جديد كما هو الحال الكود التالي:

```
new Date();
```

بالطبع أنت الآن تنظر إلى الـ Date() على أنها دالة لآنها تأخذ نفس شكل الدالة ولكن Date هي اسم لفئة التاريخ والوقت فقد تعلمت في السابق أن الدالة لا يبدأ اسمها بحرف كبير إذاً لماذا وضعنا أقواس الملحقات ..!؟

في الواقع هذا يرجع لطبيعة إنشاء بنية الفئة من الأساس ولكي تستوعب ما أقول تابع الكود التالي الذي يمثل الهيكل الأساسي لأي فئة:

```
package my.package{
    class MyClass {
        var myProperty:Number;
        function MyClass() {
        .........
        }
        function myMethod() {
        .........
        }
    }
}
```

أي فئة هي جزء من حزمة package ولذلك فكل حزمة تشتمل على عدة فئات ويمكن أن يكون للحزمة اسماً فيُكتب الحرف الأول من كل مقطع صغيراً وإذا كان الاسم عبارة عن عدة مقاطع تفصل بنقطة ,مجرد عُرف في التسمية, ومثال على ذلك هناك حزمة في الأكشن إسكربت اسمها flash.display وهي تشتمل على عدة فئات من ضمنها فئة الـ MovieClip, في المثال السابق هناك حزمة اسمها my.package.

وكما قولنا في السابق أن كل فئة تحتوي على مجموعة خصائص (متغيرات) وآليات. في المثال السابق هناك فئة اسمها MyClass تحتوي على خاصية اسمها myProperty وهناك دالة تحمل نفس اسم الفئة هذه الدالة تسمى دالة التكوين. دالة التكوين هي نوع خاص من الدوال (الآليات) داخل كل الفئات وما تحويه هذه الدالة من أوامر يتم تنفيذه تلقائياً بدون استدعاء بمجرد إنشاء كائن جديد من هذه الفئة وقد تحتاج دالة التكوين إلى معاملات ولذلك لابد من نعطي لهذه المعاملات قيمة عند تكوين كائن جديد. ويتم ذلك عن طريق كتابة قيمة هذه المعاملات بين اقواس ملحقات بعد اسم الفئة أثناء إنشاء كائن جديد.

فمثلاً لو أن لدينا فئة خاصة بالمدرس وأريد عندما يُكون كائن جديد من هذه الفئة أن يُحدد المرتب الذي سيأخذه. ولذلك كل من يقوم بإنشاء كائن جديد من هذه الفئة عليه ان يكتب قيمة للمرتب بين أقواس الملحقات.

في المثال السابق دالة التكوين كان اسمها تماماً مثل الفئة MyClass حتي في عرف التسمية. وهناك آلية (دالة) اسمها MyMethod.

هذه جزيئة متقدمة يمكنك تجاهلها الآن ولكن قصدت أن تعرف لماذا نضع أقواس ملحقات. فأقواس الملحقات تحتوي على قيم معاملات تحتاجها دالة التكوين.

ونعود للكود الرئيسي الذي نتعامل معه وهو:

```
new Date();
```

وبما أن دالة التكوين لفئة الـ Date لا تحتاج معاملات إذاً يمكن كتابتها كالآتي:

```
new Date;
```

ولكن جرت العادة على أنه حتى إذا لما تكن دالة التكوين تحتاج معاملات فنقوم وضع أقواس الملحقات فارغة.

بهذه الصورة السابقة لن تستطيع التعامل مع هذا الكائن الجديد. ببساطة لأنه لا يمتلك اسماً لكي تستطيع التحدث إليه من خلاله. ولذلك علينا إعطائه اسماً وكما تعلمت في لغة الأكشن إسكربت فإن كلمة اسم تساوي متغير إذاً علينا إنشاء متغير.

قد رأيت في السابق كيف قمنا بإنشاء متغير داخل أقواس ملحقات الدالة عن طريق كتابة اسمه وتحديد الفئة التي ينتمي إليها الكائن صاحب هذا الاسم. أما للإنشاء متغير خارج نطاق الدالة نقوم بكتابة كلمة var ثم كتابة الاسم الذي نريده ثم نحدده الفئة التي ينتمي إليها الكائن صاحب هذا الاسم ثم نعطيه لما نشاء. ليصبح الكود كالاتي:

```
var curDate:Date = new Date();
```

كما تري فقد كتابنا كلمة var, وهي إختصار لكلمة Variable أى متغير, ثم حددنا الاسم الذي نريد ثم وضعنا نقطتين لكي نحدد الفئة التي ينتمي إليها الكائن صاحب هذا الاسم ثم أعطينا هذا الاسم إلى الكائن الذي أنشأنه.

إذاً المعنى المقصود أننا نقول للآكشن إسكريبت أن الاسم curDate سيكون اسم لكائن من فئة التاريخ والوقت Date وسنعطي هذا الاسم لكائن جديد من هذه الفئة يمثل الوقت والتاريخ الحالي.

2.المصـــفوفات:

المصفوفة هي نوع خاص من المتغيرات. مع إختلاف أن المصفوفات عملية تسمية لمجموعة قيم وليس قيمة واحدة. ولتوضيح الأمر تخيل المثال التالي:

لو تتذكر أيام الدراسة في المدرسة كانت هناك عدة فصول ولكل فصل قائمة تضم (تصف, ترتب) اسماء عدد من الطلاب المتواجدين في هذا الفصل بالترتيب ولكل طالب رقمه الخاص في هذه القائمة طوال السنة, وهذه هي فكرة المصفوفة. فكل القائمة هي مصفوفة اسم الفصل يمثل اسم المصفوفة مثلاً قائمة فصل 1/3 وكل اسم طالب في القائمة يمثل عنصر من عناصر المصفوفة.

ولنطبق هذا المثال من منظور الأكشن إسكريبت ..!

فالمصفوفة كما قلت ما هي إلا عملية تسمية لمجموعة قيم, أي قيم, فممكن أن تحتوي على أعداد وقيم نصية ومتغيرات ومصفوفات أخري. المصفوفة عبارة عن عدة أمكان ولكل مكان رقم محدد. بمعنى أني أعطي لكل قيمة رقم محدد داخل المصفوفة ولكي أشير إلى قيمة ما داخل المصفوفة فإني أشير إلى رقم المكان المسند إلى القيمة التي أريدها. والترتيب يبدأ من الصفر بمعنى أن المكان الأول رقمه 0 ثم الذي يليه 1 ثم 2 وهكذا. وكالعادة الأكشن إسكريبت فأي مصفوفة تنتمي إلى فئة المصفوفات Array Class.

لكي يتضح لك الأمر تابع معي المثال التالي الذي ننشىء فيه مصفوفة تضم اسماء أيام الأسبوع:

```
var arWeek:Array = new Array ();
arWeek[0] = "الأحد";
arWeek[1] = "الإثنين";
arWeek[2] = "الثلاثاء";
arWeek[3] = "الأربعاء";
arWeek[4] = "الخميس";
arWeek[5] = "الجمعة";
arWeek[6] = "السبت";
```

بالنسبة لشرح الكود السابق:

- بما أن المصفوفات هي نوع خاص من المتغيرات ولذلك فكما رأيت قمنا بعمل متغير جديد واسميناه arWeek وحددنا الفئة التي ينتمي إليها الكائن صاحب هذا الاسم وهي Array واعطينا هذا الاسم لكائن جديد من فئة المصفوفات.
- باقي الكود قمنا فيه بملىء المصفوفة بالعناصر, وهي عبارة عن قيم نصية, عن طريق تحديد رقم مكان لكل قيمة. وذلك بإستعمال الأقواس المستقيمة [..] هذه الأقواس المستقيمة تسمى عامةً في لغات البرمجة أقواس التفصيل. فكما قُلت أن المصفوفة هي اسم لمجموعة قيم ولكل قيم رقم معين داخل هذه المصفوفة. إذاً بإستعمال أقواس التفصيل نقوم بتحديد رقم المكان مثل arWeek[0] ثم نسنده لقيمة معينة.
- عندما نحتاج إلى الإشارة مثلاً ليوم الجمعة في الكود فإننا نكتب arWeek[5] ويقوم الأكشن إسكربت بإستبدال هذا التعبير بالقيمة الحقيقة وهي الجمعة.

3.التعلـيقات:

ربما تحتاج إلى إضافة تعليقات إلى الكود الذي تكتبه لمزيد من التفصيل لك أو أثناء شرحك الكود للأحد ويمكنك كتابته باللغة العربية إن أردت. وهناك طريقتين لكتابة التعليقات:

الأولى: في حالة إذا كان التعليق عبارة عن سطر واحد بإضافة شرطتين مائلتين // مثال:

```
//هذا مثال على تعليق عبارة عن سطر واحد
```

الثانية: في حالة إذا كان التعليق عبارة عن عدة اسطر بإضافة /* في بداية التعليق و */ في نهاية التعليق مثال:

```
/* هذا مثال على تعليق كبير
عبارة عن عدة اسطر*/
```

والجدير بالذكر أن التعليق يأخذ اللون الرمادي داخل الكود.

انتهـــــــــــــــــى !